

```

1<--`day` 02 함수
2 /*단일 행 함수, 그룹 함수*/
3
4 SELECT *
5 FROMEMPLOYEE;
6
7 -- SUBSTR() : 지정한 위치에서부터 지정한 개수 만큼의 문자열 반환
8 -- SUBSTR(테이블, 인덱스번지&방향, 문자 갯수)
9 /*0 은 방향을 의미
10 0보다 크면 양수값, 양수의 인덱싱 (1의 의미를 포함)
11 변수의 타입은 왼쪽 정렬시 문자열, 오른쪽 정렬시 숫자*/
12
13 SELECTEMP_NAME,
14 LENGTH(SUBSTR(EMP_NO, 0,6)) /*함수 중첩 가능*/
15 FROMEMPLOYEE;
16
17 -- DUMMY TABLE(DUAL)
18 -- 인덱스가 1부터 시작
19 /*함수는 SELECT, WHERE절에서 사용 가능*/
20
21 SELECTSUBSTR('THIS IS A SIAT', -4,4)
22 FROMDUAL; /*FROM절 대상테이블에 DUMMY 넣을 수도 있음*/
23
24 -- SYSDATE: 오늘 날짜를 반환하는 함수
25
26 SELECTSYSDATE
27 FROMDUAL;
28
29 -- ADD_MONTHS() : 개월 수를 더해주는 함수
30 SELECTEMP_NAME,
31 HIRE_DATE,
32 HIRE_DATE +1, /* 일수를 더해줌 */
33 ADD_MONTHS(HIRE_DATE, 2)
34 FROMEMPLOYEE;
35
36 -- 요구사항)
37 -- 입사일을 기준으로 근속연수가 20년일 되는 일자를 조회한다면?
38
39 SELECTEMP_NAME,
40 HIRE_DATE ,
41 ADD_MONTHS(HIRE_DATE,240)
42 FROMEMPLOYEE;
43
44
45 SELECTMONTHS_BETWEEN('25/03/24','25/01/01') /*개월 수 차이*/
46 FROMDUAL;
47
48 /*결과값은 음수로 반환*/
49 SELECTMONTHS_BETWEEN('25/01/01', '25/03/24') /*개월 수 차이*/

```

```

50 FROM DUAL;
51
52 -- 요구사항)
53 -- 오늘 날짜를 기준으로 입사한지 20년이 넘는 직원의 이름, 입사일, 근무연수 조회한다면?
54
55 SELECT *
56 FROM EMPLOYEE;
57
58 SELECT EMP_NAME AS "이름",
59        HIRE_DATE AS "입사일",
60        TRUNC(MONTHS_BETWEEN(SYSDATE, HIRE_DATE)/12, 0) AS "근무연수"
61 FROM EMPLOYEE
62 WHERE MONTHS_BETWEEN(SYSDATE, HIRE_DATE) > 300;
63
64 -- ROUND (반올림), TRUNC (절삭)
65 /*양수일 경우 실수자릿수를 의미, 음수일 경우 정수 자릿수 2자리를 의미) */
66 SELECT ROUND(123.315, 2), TRUNC(123.315)
67 FROM DUAL;
68
69 -- 그룹 함수 (AVG, MEAN)
70 -- 그룹 함수를 SELECT절에서 사용할 경우, 일반 컬럼은 정의될 수 없다.
71
72 SELECT AVG(SALARY)
73 FROM EMPLOYEE;
74
75
76 -- 데이터 타입변환 함수
77
78 /*
79 TO_NUMBER()
80 TO_CHAR()
81 TO_DATE()
82
83 TO_CHAR() TO_DATE()
84 NUMBER -----> CHARACTER-----> DATE
85 <----- <-----
86 TO_NUMBER() TO_CHAR()
87 */
88
89 SELECT 250324,
90        MONTHS_BETWEEN('25/01/01', TO_DATE(TO_CHAR(250324)))
91 FROM DUAL;
92
93 /*형변환: 묵시적 캐스팅, 명시적 캐스팅*/
94
95 SELECT 1234,
96        TO_CHAR(1234, 'L9999'), /*포매팅. L은 LOCAL, 9는 자릿수를 의미*/
97        TO_CHAR(1234, '$9999'),
98        TO_CHAR(1234, 'L999')

```

```

99<---\ 25 23 24
FROMDUAL;
100
101-- DATE -> CHAR
102/*
103FORMAT
104- YYYY/ YY
105- MONTH / MON / MM / RM
106- HH MI SS (시, 분, 초)
107- AM | PM
108 * * */
109
110
111SELECT SYSDATE,
112 TO_CHAR(SYSDATE, 'YY'),
113 TO_CHAR(SYSDATE, 'YEAR'),
114 TO_CHAR(SYSDATE, 'MM'),
115 TO_CHAR(SYSDATE, 'MONTH'),
116 TO_CHAR(SYSDATE, 'RM'),
117 TO_CHAR(SYSDATE, 'DD'), /*일*/
118 TO_CHAR(SYSDATE, 'DY'), /*요일*/
119 TO_CHAR(SYSDATE, 'Q'), /*분기*/
120 TO_CHAR(SYSDATE, 'AM HH')
121FROMDUAL;
122
123
124/*문자열 데이터로 반환하여 연, 월, 일만 추출*/
125SELECTEMP_NAME,
126 HIRE_DATE,
127 TO_CHAR(HIRE_DATE, 'YY/MM/DD'),
128 TO_CHAR(HIRE_DATE, 'YY "년 " MM "월 " DD "일" ')
129FROMEMPLOYEE;
130
131SELECTEMP_NAME,
132 HIRE_DATE,
133 SUBSTR(HIRE_DATE, 1, 2) || '년' ||
134 SUBSTR(HIRE_DATE, 4, 2) || '월' ||
135 SUBSTR(HIRE_DATE, 7, 2) || '일'
136FROMEMPLOYEE;
137
138
139/* Q. ' ' 와 ""차이 ? */
140
141SELECTEMP_NO,
142 SUBSTR(EMP_NO, 1, 6),
143 SUBSTR(EMP_NO, 8, 7),
144 SUBSTR(EMP_NO, 1, 6) + SUBSTR(EMP_NO, 8, 7) /*문자열 정수 + 문자열
145FROMEMPLOYEE;
146

```

```

147 -- 기타 함수
148 -- NVL() : 널값을 다른 값으로 대체
149 -- 직원의 이름, 급여, 연봉을 조회하고 싶다면?
150
151 SELECT *
152 FROM EMPLOYEE;
153
154 SELECT EMP_NO,
155        SALARY,
156        (SALARY*12) + ((SALARY*12)*NVL(BONUS_PCT,0)) /*보너스 금액 포함한
157 FROM EMPLOYEE
158 WHERE SALARY > 3500000;
159
160 -- DECODE(EXPR, [SEARCH RESULT], DEFAULT) : 오라클 전용 함수
161 -- ANSI : 벤더사와 상관없이 사용하는 표준함수
162 -- CASE EXPR [WHEN SEARCH THEN RESULT] ELSE DEFAULT END; 정확하게
163 -- CASE [WHEN CONDITION THEN RESULT] ELSE DEFAULT END; 데이터
164 -- IF ~ ELSE 논리를 제한적으로 구현하는 오라클 함수
165
166
167 SELECT EMP_NAME,
168        EMP_NO,
169        DECODE(SUBSTR(EMP_NO,8,1),
170        '1','남', /*SEARCH RESULT*/
171        '2','여') AS 성별 /*DEFAULT*/
172 FROM EMPLOYEE
173 WHERE DEPT_ID = '50' AND
174        SUBSTR(EMP_NO, 8,1) = '2';
175
176 /*
177 요구사항)
178 - 급여를 인상하고 싶다
179 - 부서별(50: 10%, 80: 15%, 90: 20%) 차등하여 급여를 인상
180 - 나머지 부서의 직업의 급여는 동결
181 */
182
183 SELECT *
184 FROM DEPARTMENT
185
186 SELECT EMP_NAME,
187        DEPT_ID,
188        SALARY,
189        DECODE(DEPT_ID, /*논리적 제한 | EXPR*/
190        '50', SALARY*1.1, /*SEARCH RESULT*/
191        '80', SALARY*1.15, /*SEARCH RESULT*/
192        '90', SALARY*1.2, /*SEARCH RESULT*/
193        SALARY) AS "인상급여" /*ELSE*/
194 FROM EMPLOYEE
195

```

```

196
197 /*CASE END 구문*/
198 /*EXACTLY 비교*/
199 -- CASE EXPR [WHEN SEARCH THEN RESULT] ELSE DEFAULT END;
200
201 SELECTEMP_NAME,
202 DEPT_ID,
203 SALARY,
204 CASE DEPT_ID
205     WHEN '50' THEN SALARY*1.1
206     WHEN '80' THEN SALARY*1.15
207     WHEN '90' THEN SALARY*1.2
208     ELSE SALARY
209 END AS "인상급여"
210 FROMEMPLOYEE;
211
212 /*조건식*/
213 -- CASE [WHEN CONDITION THEN RESULT] ELSE DEFAULT END;
214 /*
215 급여가 3000000 이하면 초급
216 4000000 이하면 중급
217 3000000 초과면 고급
218 */
219
220 SELECTEMP_NAME,
221 SALARY,
222 CASE
223     WHEN SALARY <=3000000 THEN '초급'
224     WHEN SALARY <=4000000 THEN '중급'
225     ELSE '고급'
226 END AS "급여 등급"
227 FROMEMPLOYEE;
228
229
230 -- 그룹함수 (SUM, AVG, MIN, MAX, COUNT)
231 -- 주의사항) SELECT절에 그룹함수를 사용하면 일반컬럼은 사용할 수 없다.
232
233 SELECTCOUNT(*), MIN(SALARY), MAX(SALARY), AVG(SALARY),
234 FROMEMPLOYEE;
235
236 -- ORDER BY 절: 특정한 컬럼값을 기준으로 정렬할 때. SELECT문에서 마지막에 사용
237 -- ORDER BY 기준컬럼 [ ASC(오름차순) | DESC(내림차순) ] 기준컬럼 [ ASC
238 -- 컬럼이름, 컬럼 별칭, 컬럼 기술 순서로 표현
239
240 /* 요구사항) 부서번호가 50번인 사원의 이름과 급여를 조회하라.*/
241
242 SELECTEMP_NAME AS N,
243 SALARY AS S
244 FROMEMPLOYEE

```

```

245 WHERE DEPT_ID = '50';
246 ORDER BY 2 DESC;
247
248
249 /*[Additional SELECT - 함수]
250 1. 영어영문학과 (학과코드 002) 학생들의 학번과 이름, 입학 년도를 입학 년도가 빠른
251 순으로 표시하는 SQL 문장을 작성하시오. ( 단, 헤더는 "학번", "이름", "입학년도" 가
252 표시되도록 한다. ) */
253
254 SELECT STUDENT_NO AS "학번",
255 STUDENT_NAME AS "이름",
256 ENTRANCE_DATE AS "입학년도"
257 FROM TB_STUDENT
258 WHERE DEPARTMENT_NO = 2
259 ORDER BY ENTRANCE_DATE;
260
261 /*2. 춘 기술대학교의 교수 중 이름이 세 글자가 아닌 교수가 한 명 있다고 한다. 그 교수의
262 이름과 주민번호를 화면에 출력하는 SQL 문장을 작성해 보자. (* 이때 올바르게 작성한 SQL
263 문장의 결과 값이 예상과 다르게 나올 수 있다. 원인이 무엇일지 생각해볼 것) */
264
265
266 SELECT PROFESSOR_NAME, PROFESSOR_SSN
267 FROM TB_PROFESSOR
268 WHERE LENGTH (PROFESSOR_NAME) >3 OR LENGTH (PROFESSOR_NAME) <3 ;
269
270
271 /*3. 춘 기술대학교의 남자 교수들의 이름과 나이를 출력하는 SQL 문장을 작성하시오. 단
272 이때 나이가 적은 사람에서 많은 사람 순서로 화면에 출력되도록 만드시오. (단, 교수 중
273 2000 년 이후 출생자는 없으며 출력 헤더는 "교수이름", "나이"로 한다. 나이는 '만' 으로
274 계산한다. ) */
275
276 SELECT *
277 FROM TB_PROFESSOR;
278
279 SELECT PROFESSOR_NAME AS "교수이름",
280 TO_NUMBER (TO_CHAR (SYSDATE, 'YYYY'))
281 -
282 TO_NUMBER (19 || SUBSTR (PROFESSOR_SSN, 1, 2)) AS "나이"
283 FROM TB_PROFESSOR
284 WHERE SUBSTR (PROFESSOR_SSN, 8, 1) = '1'
285 ORDER BY 2 ASC , 1 ASC ; /*인덱스 번지 혹은 별칭*/
286
287 /*4. 교수들의 이름 중 성을 제외한 이름만 출력하는 SQL 문장을 작성하시오. 출력 헤더는
288 '이름' 이 찍히도록 한다. (성이 2 자인 경우는 교수는 없다고 가정하시오) */
289
290 SELECT SUBSTR (PROFESSOR_NAME, 2)
291 AS "이름"
292 FROM TB_PROFESSOR

```

```

293< 19 25 02 04 2005년 2월 27일 목요일 오후 4:30
294
295 /*5. 춘 기술대학교의 재수생 입학자를 구하려고 한다. 어떻게 찾아낼 것인가? 이때,
296 19 살에 입학하면 재수를 하지 않은 것으로 간주한다.*/
297
298 SELECT*
299 FROMTB_STUDENT;
300
301 SELECTSTUDENT_NO,
302 STUDENT_NAME
303 FROM TB_STUDENT
304 WHERE TO_NUMBER(TO_CHAR(ENTRANCE_DATE, 'YYYY'))
305 -
306 TO_NUMBER(19||SUBSTR(STUDENT_SSN,1,2)) > 19;
307
308
309
310 /*6. 2020 년 크리스마스는 무슨 요일인가?*/
311
312 SELECTTO_CHAR(TO_DATE('20/12/25'),'DY')
313 FROMDUAL;
314
315
316 /*7. TO_DATE('99/10/11','YY/MM/DD'), TO_DATE
317 월 몇 일을 의미할까? 또 TO_DATE('99/10/11','RR/MM/DD'),
318 TO_DATE('49/10/11','RR/MM/DD') 은 각각 몇 년 몇 월 몇 일을 의미할까?*/
319
320 SELECTTO_CHAR(TO_DATE('99/10/11'),'YY "년 " MM "월 " DD "일"
321 TO_CHAR(TO_DATE('49/10/11'),'YY "년 " MM "월 " DD "일" ')
322 FROMDUAL;
323
324
325
326 /*8. 춘 기술대학교의 2000 년도 이후 입학자들은 학번이 A 로 시작하게 되어있다. 2000
327 이전 학번을 받은 학생들의 학번과 이름을 보여주는 SQL 문장을 작성하시오.*/
328
329 SELECT*
330 FROMTB_STUDENT
331
332 SELECTSTUDENT_NO, STUDENT_NAME
333 FROMTB_STUDENT
334 WHERE TO_NUMBER(TO_CHAR(ENTRANCE_DATE, 'YYYY')) < 2000;
335
336
337 /*9. 학번이 A517178 인 한아름 학생의 학점 총 평점을 구하는 SQL 문을 작성하시오.
338 이때 출력 화면의 헤더는 "평점" 이라고 찍히게 하고, 점수는 반올림하여 소수점 이하 한
339 자리까지만 표시한다.*/
340
341 SELECT*

```

```

342 FROM TB_GRADE tg
343
344 SELECT ROUND (AVG (POINT), 1) AS "평점"
345 FROM TB_GRADE
346 WHERE STUDENT_NO = 'A517178';
347
348
349 /*10. 학과별 학생수를 구하여 "학과번호", "학생수 (명)" 의 형태로 헤더를 만들어
350 출력되도록 하시오.*/
351
352 SELECT *
353 FROM TB_STUDENT;
354
355 SELECT DEPARTMENT_NO AS "학과번호",
356        COUNT (*) AS "학생수 (명)"
357 FROM TB_STUDENT
358 GROUP BY DEPARTMENT_NO
359 ORDER BY 1 ASC;
360
361
362
363 /*11. 지도 교수를 배정받지 못한 학생의 수는 몇 명 정도 되는 알아내는 SQL 문을
364 작성하시오.*/
365
366
367 SELECT COUNT (*)
368 FROM TB_STUDENT
369 WHERE COACH_PROFESSOR_NO IS NULL;
370
371
372 -----
373
374 /*12. 학번이 A112113 인 김고운 학생의 년도 별 평점을 구하는 SQL 문을 작성하시오.
375 이때 출력 화면의 헤더는 "년도", "년도 별 평점" 이라고 찍히게 하고, 점수는 반올림하여
376 소수점 이하 한 자리까지만 표시한다.*/
377
378 SELECT *
379 FROM TB_GRADE
380 WHERE STUDENT_NO = 'A112113';
381
382
383
384 SELECT SUBSTR (TERM_NO, 0, 4) AS "년도",
385        ROUND (AVG (POINT), 1) AS "년도 별 평점"
386 FROM TB_GRADE
387 WHERE STUDENT_NO = 'A112113'
388 GROUP BY SUBSTR (TERM_NO, 0, 4);
389

```



```

390
391 /*13. 학과 별 휴학생 수를 파악하고자 한다. 학과 번호와 휴학생 수를 표시하는 SQL
392 작성하시오.*/
393
394 SELECT *
395 FROM TB_STUDENT;
396
397
398 SELECT DEPARTMENT_NO AS "학과코드명",
399        COUNT(CASE ABSENCE_YN /*집계함수는 표현식 사용 가능*/
400              WHEN 'Y' THEN '1'
401              WHEN 'N' THEN NULL
402              END) AS "휴학생 수"
403 FROM TB_STUDENT
404 GROUP BY DEPARTMENT_NO
405 ORDER BY 1;
406
407 /*강사님 풀이*/
408 SELECT DEPARTMENT_NO AS "학과코드명",
409        SUM (CASE ABSENCE_YN /*집계함수는 표현식 사용 가능*/
410              WHEN 'Y' THEN 1
411              ELSE 0 END) AS "휴학생 수"
412 FROM TB_STUDENT
413 GROUP BY DEPARTMENT_NO
414 ORDER BY 1;
415
416
417
418
419 /*14. 춘 대학교에 다니는 동명이인 (同名異人) 학생들의 이름을 찾고자 한다. 어떤 SQL
420 문장을 사용하면 가능하겠는가?*/
421
422 SELECT STUDENT_NAME AS "동일이름",
423        COUNT(STUDENT_NO) AS "동명인 수"
424 FROM TB_STUDENT
425 GROUP BY STUDENT_NAME
426 HAVING COUNT(STUDENT_NO) > 1
427 ORDER BY STUDENT_NAME;
428
429
430
431 /*15. 학번이 A112113 인 김고운 학생의 년도, 학기 별 평점과 년도 별 누적 평점 , 총
432 평점을 구하는 SQL 문을 작성하시오. (단, 평점은 소수점 1 자리까지만 반올림하여
433 표시한다.)*/
434
435 SELECT SUBSTR(TERM_NO, 1, 4) AS "연도",
436        SUBSTR(TERM_NO, 5, 2) AS "학기",
437        ROUND(AVG(POINT), 1) AS "평점"

```

```
438<--\ ^5 ^3 ^4
439FROM TB_GRADE
440WHERE STUDENT_NO = 'A112113'
441GROUP
442ORDER BY1;
```